BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images,
Please do not report the images to the
Image Problem Mailbox.

	Search Text	DBs
1	graphic with manag$4 with (fifo or memory)	USPAT; EPO; JPO
2	graphic with manag$4 with (fifo)	USPAT; EPO; JPO
3	(graphic with manag$4 with memory) same fifo	USPAT; EPO; JPO
4	(contoller with manag$4 with memory) same fifo	USPAT; EPO; JPO
5	(controller with manag$4 with memory) same fifo	USPAT; EPO; JPO
6	controller with manag$4 with fifo	USPAT; EPO; JPO
7	controller with manag$4 with fifo with memory	USPAT; EPO; JPO
8	graphic with manag$4 with memory	USPAT; EPO; JPO
9	(graphic adj (contoller or engine)) with manag$4 with memory	USPAT; EPO; JPO
10	(graphic adj (contoller or engine)) with memory	USPAT; EPO; JPO
11	((graphic adj (contoller or engine)) with memory) same fifo	USPAT; EPO; JPO
12	((graphic adj (contoller or engine)) with memory) same calculat$4	USPAT; EPO; JPO
13	((graphic adj (contoller or engine)) with memory) and (calculat$4 same time same fill)	USPAT; EPO; JPO
14	(graphic adj (contoller or engine)) and (calculat$4 same time same fill same buffer)	USPAT; EPO; JPO
15	(graphic adj (contoller or engine)) and (calculat$4 with time with fill with buffer)	USPAT; EPO; JPO
16	calculat$4 with time with fill with buffer	USPAT; EPO; JPO
17	6173381.pn.	USPAT; EPO; JPO

DOCUMENT-IDENTIFIER: US 6173381 B1
TITLE: Memory controller including embedded data compr[illegible] and decompression engines

KWIC

BSPR:
The execution engine in turn couples to a graphics engine which couples through FIFO buffers to one or more symmetrical memory control units. The graphics engine is similar in function to graphics processors in conventional computer systems and includes line and triangle rendering operations as well as span line interpolators. An instruction storage/decode block is coupled to the bus interface logic which stores instructions for the graphics engine and memory compression/decompression engines. A Window Assembler is coupled to the one or more memory control units. The Window Assembler in turn couples to a display storage buffer and then to a display memory shifter. The display memory shifter couples to separate digital to analog converters (DACs) which provide the RGB signals and the synchronization signal outputs to the display monitor. The window assembler includes a novel display list-based method of assembling pixel data on the screen during screen refresh, thereby improving system performance. In addition, a novel antialiasing method is applied to the video data as the data is transferred from system memory to the display screen. The internal graphics pipeline of the IMC is optimized for high end 2D and 3D graphical display operations, as well as audio operations, and all data is subject to operation within the execution engine and/or the graphics engine as it travels through the data path of the IMC.

DEPR:
The graphics engine 212 couples to respective memory control units referred to as memory control unit #1220 and memory control unit #2222 via respective FIFO buffers 214 and 216, respectively. Memory control unit #1220 and memory control #2222 provide interface signals to communicate with respective banks of system memory 110. In an alternate embodiment, the IMC 140 includes a single memory control unit. The graphics engine 212 reads graphical data from system

	U		Document ID	Issue Dat	Current	Title
1			US 6370631 B	20020409	711/170	Memory controller including compression/
2			US 6173381 B	20010109	711/170	Memory controller including embedded da
3			US 6170047 B	20010102	711/170	System and method for managing system
4			US 5623634 A	19970422	711/151	Resource allocation with parameter count
5			US 6226695 B	20010501	710/5	Information handling system including non-

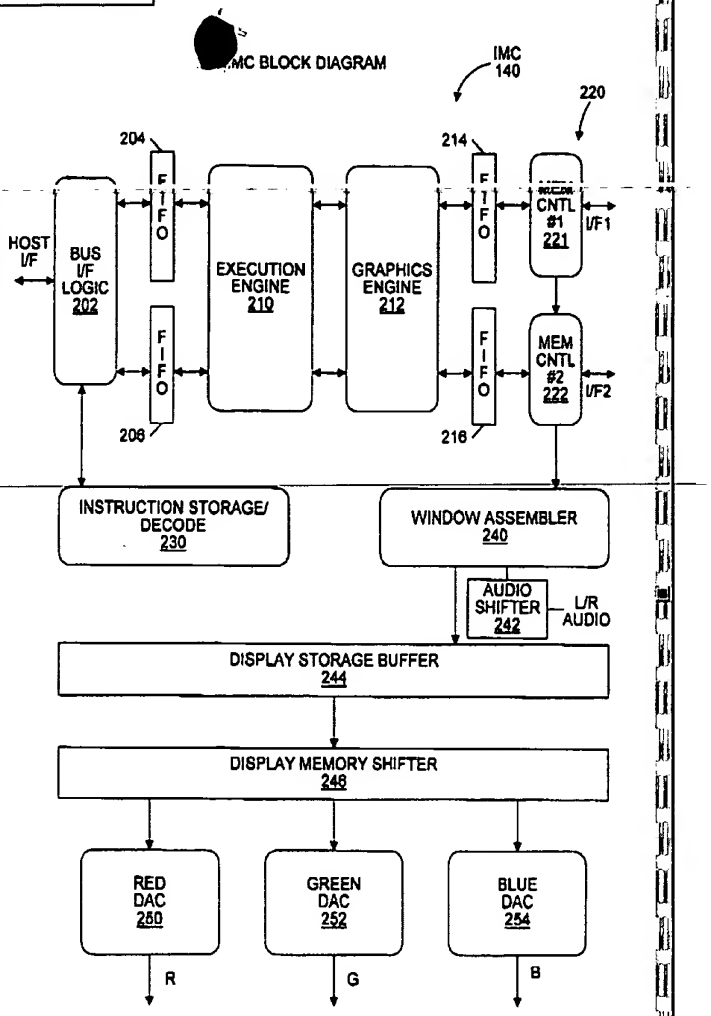


FIG. 5

DOCUMENT-IDENTIFIER: US 5805905 A
TITLE: Method and apparatus for arbitrating requests at two or more levels of priority using a single request line

DEPR:
The conditions under which the device 914 makes a request for control of the shared resource 910 are not important for an understanding of the broadest aspects of the present invention. In an embodiment in which the device 914 is a video controller having a graphics engine and a FIFO for buffering pixel data for display, and the shared resource 910 is a memory, such a request can be made whenever either the graphics engine requires access to the memory or the FIFO empties to a predefined high water mark, such as 3/4 full. Similarly, the conditions under which the device 914 increases the level of its request to a high priority request, are also unimportant for an understanding of the broadest aspects of the invention. Again, however, in an embodiment in which the device 914 is a video controller having a FIFO for buffering pixel data to a display, the device 914 may do so when the FIFO empties to a predefined low water mark level, such as 1/4 full.

DEPR:
The video controller 126 includes, among other things, two primary elements: a FIFO 218 (including its control circuitry), and a graphics engine (GE) 220. The graphics engine 220 reads from and writes to the display memory portion of unified memory 118 as necessary to follow various graphics commands received over the PCI bus 124, and the FIFO 218 reads data from the frame buffer portion of the unified memory 118 for output to the display 128 (FIG. 1).

DEPR:
The local arbiter 232 arbitrates between requests for access to the unified memory 118 from the FIFO 218 and the graphics engine 220. It receives a FIFO high water mark (fhwm#) signal and a FIFO low water mark (flwm#) signal from the FIFO 218, and a graphics engine request (ger#) signal from the graphics

	U	Document ID	Issue Dat	Current	Title
3		US 6170047 B	20010102	711/170	System and method for managing system
4		US 5623634 A	19970422	711/151	Resource allocation with parameter count
5		US 6226695 B	20010501	710/5	Information handling system including non
6		US 6282588 B	20010828	710/27	Data transfer method and device
7		US 5805905 A	19980908	710/244	Method and apparatus for arbitrating requ

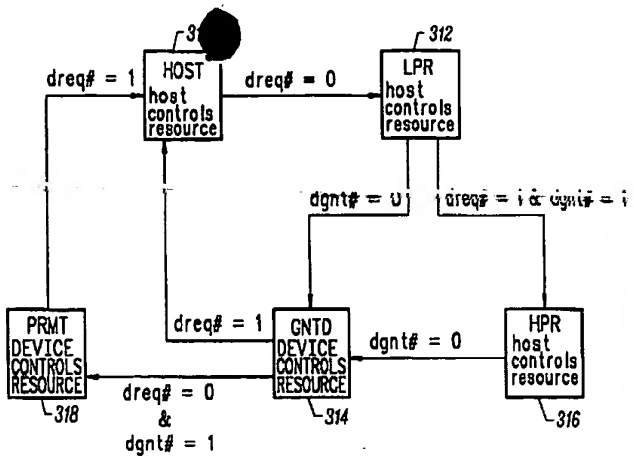


FIG. 3

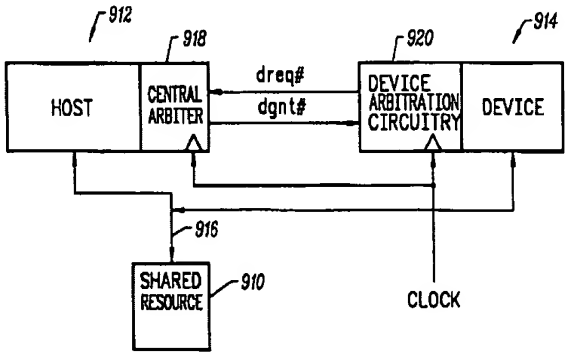


FIG. 9

DOCUMENT-IDENTIFIER: US 5838334 A
TITLE: Memory and graphics controller which performs po[illegible]ased display list video refresh operations

KWIC

BSPR:
The integrated memory controller of the preferred embodiment includes a bus interface unit which couples through FIFO buffers to an Execution Engine. The Execution Engine preferably includes a digital signal processor (DSP) core which performs compression and decompression operations, as well as texture mapping, and which also assembles display refresh lists according to the present invention. The Execution Engine in turn couples to a Graphics Engine which couples through FIFO buffers to one or more symmetrical memory control units. The Graphics Engine is similar in function to graphics processors in conventional computer systems and includes line and triangle rendering operations as well as span line interpolators. An instruction storage/decode block is coupled to the bus interface logic which stores instructions for the Graphics Engine and the Execution Engine.

DEPR:
The Graphics Engine 212 couples through respective FIFO buffers 214 and 216 to a memory control unit 220, wherein the memory control unit 220 comprises respective memory control units referred to as memory control unit #1 221 and memory control unit #2 222. The FIFO buffers 214 and 216 couple to the memory control units 221 and 222, respectively. Memory control unit #1 221 and memory control #2 222 provide interface signals to communicate with respective banks of system memory 110. In an alternate embodiment, the IMC 140 includes a single memory control unit. The Graphics Engine 212 reads graphical data from system memory 110, performs various graphical operations on the data, such as formatting the data to the correct x,y addressing, and writes the data back to system memory 110. The Graphics Engine 212 performs operations on data in the system memory 110 under CPU control using the high level graphical protocol of the present invention. In many instances, the Graphics Engine 212 manipulates

	U	Document ID	Issue Dat	Current	Title
16		US 5854638 A	19981229	345/542	Unified memory architecture with parallel
17		US 6067098 A	20000523	345/531	Video/graphics controller which performs
18		US 5678036 A	19971014	345/522	Graphics system and method for minimizi
19		US 5715437 A	19980203	345/519	System for, and method of, processing in
20		US 5838334 A	19981117	345/503	Memory and graphics controller which per

[45] Date of Patent: Nov. 17, 1998

[54] MEMORY AND GRAPHICS [illegible]CONTROLLER WHICH PERFORMS POINTE[illegible]SED DISPLAY LIST VIDEO REFRESH OPERATIONS

[76] Inventor: Thomas A. Dye, 6621 Candle Ridge Cove, Austin, Tex. 78731

[21] Appl. No.: 565,103

[22] Filed: Nov. 30, 1995

Related U.S. Application Data

[63] Continuation of Ser. No. 522,129, Aug. 31, 1995, abandoned, Continuation-in-part of Ser. No. 340,667, Nov. 16, 1994.

[51] Int. Cl.[6] G06F 15/16

[52] U.S. Cl. 345/503; 345/501; 345/507; 345/525; 345/339

[58] Field of Search 395/501, 503, 395/507, 509, 515, 516, 520–522, 339–345, 357, 186, 510, 203, 525; 345/112–114, 119–121, 186, 187, 189, 190, 200, 203

[56] References Cited

U.S. PATENT DOCUMENTS

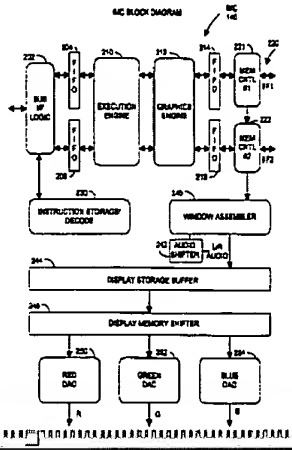
4,945,499	7/1990	Asari et al.	395/522
4,967,375	10/1990	Pelham et al.	395/515
5,097,411	3/1992	Doyle et al.	395/522
5,321,805	6/1994	Hayman et al.	395/502
5,502,462	3/1996	Mical et al.	395/185

Primary Examiner—Mathew M. Kim
Assistant Examiner—U. Chauhan
Attorney, Agent, or Firm—Conley, Rose & Tayon; Jeffrey C. Hood

[57] ABSTRACT

A graphics controller (IMC) which performs pointer-based and/or display list-based video refresh operations that enable screen refresh data to be assembled on a per window basis, thereby greatly increasing the performance of the graphical display. The graphics controller maintains pointers to various buffers in system memory comprising video or graphics display information. The graphics controller manipulates respective object information workspace memory areas corresponding to each object or window, wherein the workspace areas specify data types, color depths, 3D depth values, alpha blending information, screen position, etc. for the respective window or object on the screen. Each workspace area also includes static and dynamic pointers which point to the location in system memory where the pixel data for the respective window or object is stored. The graphics controller utilizes this information, as well as information received from the software driver regarding screen changes, to assemble a display refresh list in system memory. This information is used during the screen refresh to display the various windows or objects on the screen very quickly and efficiently. Thus, the video display can be updated with new video data without requiring any system bus data transfers, which are required in prior art computer system architectures. The graphics controller dynamically adjusts the display refresh list for movement of objects and changes in relative depth priority which appear on the display. Thus the video data for the various windows and objects is stored in respective memory areas in the system memory, and pointers assembled in the display refresh list are used to reference this data during screen updates. Therefore, data is not required to be moved in or out of a frame buffer to reflect screen changes. Rather, in many instances, either the video data for a respective window or object is changed, or only the pointers in the display refresh list are manipulated, to affect a screen change.

65 Claims, 41 Drawing Sheets



DOCUMENT-IDENTIFIER: US 5751295 A
TITLE: Graphics accelerator chip and method

------ KWIC ------

DEPR:
A FIFO graphics engine 113 is responsible for routing primitive commands and data from the PC microprocessor and memory to the graphics registers 143. The preferred FIFO graphics engine 113 contains the FIFO RAM 115, a temporary storage location 119, a state machine 121, and a multiplexer (MUX) 123. The FIFO RAM 115 provides N storage locations, M bits wide. The temporary storage location 119 is used to hold a primitive command. The temporary storage location 119 is preferably implemented as a transistor flip-flop (FF) M bits wide, but it will be recognized that other implementations may be substituted by one of ordinary skill in the art without loss of generality. The preferred state machine 121 contains the logic to implement a data flow algorithm corresponding to the graphics commands that the graphics processor is able to perform. The MUX 123 allows the routing of data from the output port 117 of the FIFO RAM 115 or the temporary storage location 119 to the graphics registers 143. A MUX 129, controlled by control line 133, is used to route data from the FIFO graphics engine data output 125 or PC data line 109 to the graphics registers 143 via data bus 131. A MUX 139, controlled by control line 133, is used to route address information from the state machine address output 135 or PC address line 111 to the graphics registers 143 via address bus 141.

	U	Document ID	Issue Dat	Current	Title
18		US 5678036 A	19971014	345/522	Graphics system and method for minimizi
19		US 5715437 A	19980203	345/519	System for, and method of, processing in
20		US 5838334 A	19981117	345/503	Memory and graphics controller which per
21		US 5774131 A	19980630	345/503	Sound generation and display control appa
22		US 5751295 A	19980512	345/503	Graphics accelerator chip and method

Beckhund et al.

[45] Date of Patent: May 12, 1998

[54] GRAPHICS ACCELERATOR CE[illegible]D METHOD

[75] Inventors: Thomas K. Beckhund, Edina; Todd C. Haug, Wyoming; Benton H. Jackson, Maplewood; David O. Sutter, Scandia; John R. Ukura, Lino Lakes, all of Minn.

[73] Assignee: Control Systems, Inc., St. Paul, Minn.

[21] Appl. No.: 429,834

[22] Filed: Apr. 27, 1995

[51] Int. Cl.[6] G06F 15/80

[52] U.S. Cl. 345/501; 345/516

[58] Field of Search 395/162, 166, 395/163, 165, 285, 126, 501, 502, 516, 521; 382/296, 297; 345/501, 502, 516, 521, 526

[56] References Cited

U.S. PATENT DOCUMENTS

4,713,779	12/1987	Graciotti et al.	395/166
5,131,080	7/1992	Hendrickson et al.	395/516
5,303,334	4/1994	Snyder et al.	395/162
5,404,445	4/1995	Matsumoto	395/162
5,469,558	11/1995	Lieberman et al.	395/285
5,517,626	5/1996	Archer et al.	395/163

OTHER PUBLICATIONS

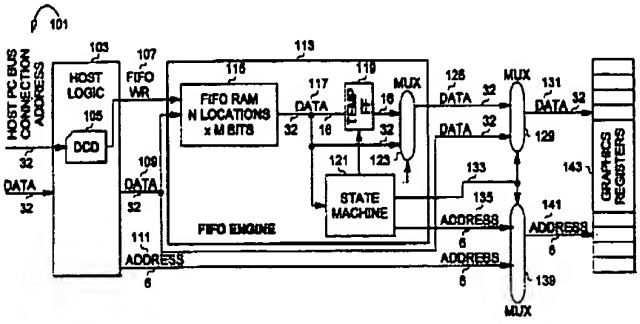
"Chapter 10 -3GA Enhanced Mode Registers." *Graphics Accelerator* Preliminary Reference Manual.

Primary Examiner—Raymond J. Bayerl
Assistant Examiner—Cao H. Nguyen
Attorney, Agent, or Firm—Merchant, Gould, Smith, Edell, Weiter & Schmidt, P.A.

[57] ABSTRACT

A graphics accelerator chip which interprets instructions and data transferred from a microprocessor via an external data bus logically coupled to the microprocessor. A host logic interface buffers the information received from the microprocessor with an on-chip first-in first-out (FIFO) memory which has an address space mapped onto a contiguous sequential address space of the microprocessor. A state machine having a temporary memory receives and interprets instructions and data from the FIFO memory, and routes them to a graphics register set which performs logical graphics operations based upon the graphics instructions and data. The temporary memory stores the last primitive command received, allowing the chip to perform multiple graphics operations where a primitive command is received from the microprocessor only once. A separate data bus from the host logic interface to the graphics register set enables direct access to the graphics registers from the microprocessor.

29 Claims, 8 Drawing Sheets



DOCUMENT-IDENTIFIER: US 6138189 A
TITLE: Network interface having adaptive transmit start po[redacted] each packet to avoid transmit underflow

----- KWIC -----

ABPL:
A network interface transmits data packets between a host computer and a network and includes a first in first out (FIFO) buffer memory with an adaptive transmit start point determined for each data packet. The network interface receives data packets from the host computer via a peripheral component interconnect (PCI) bus. A FIFO control determines the byte length of each data packet based on the header information contained in the first few received bytes of the packet. The FIFO control also measures a minimum fill time indicating the time necessary to fill the FIFO buffer memory with a predetermined minimum amount of data necessary before transmission by the FIFO buffer memory. The FIFO control calculates the time to fill the FIFO buffer memory with each packet based on the determined length and the measured minimum fill time. The time to empty the packet from the FIFO buffer memory is also calculated based upon the length of the packet and predetermined network transmission rates. If the time to empty the packet from the FIFO buffer memory is greater than or equal to the time to fill the FIFO buffer memory, the transmit start point is set to the predetermined minimum amount; otherwise, the transmit start point is adjusted in accordance with the difference in time between filling and emptying the FIFO buffer memory with the packet, a FIFO fill rate based on the measured minimum fill time, and a coefficient that accounts for latencies in the PCI bus.

	U	Document ID	Issue Dat	Current	Title
1		US 6138189 A	20001024	710/53	Network interface having adaptive transmit s
2		US 5623699 A	19970422	710/52	'Read only linear stream based cache syste
3		US 6094692 A	20000725	710/34	Network interface having adaptive transmit s
4		US 5859980 A	19990112	709/232	Network interface having adaptive transmit s
5		US 6061731 A	20000509	709/231	Read only linear stream based cache syste

[54] NETWORK INTERFACE HA[redacted]DAPTIVE TRANSMIT START POINT FO[redacted]ACH PACKET TO AVOID TRANSMI[redacted] UNDERFLOW

[75] Inventor: Mohan Kalkunte, Sunnyvale, Calif.

[73] Assignee: Advanced Micro Devices, Inc., Sunnyvale, Calif.

[21] Appl. No.: 09/098,434

[22] Filed: Jun. 17, 1998

Related U.S. Application Data

[62] Division of application No. 08/588,290, Feb. 8, 1996, Pat. No. 5,859,980.

[51] Int. Cl.[7] G06F 13/28; G06F 13/38; H04J 3/26

[52] U.S. Cl. 710/53; 370/378; 709/231; 709/232

[58] Field of Search 710/123, 35, 52, 710/34, 53; 709/231, 232, 217; 714/811; 370/253, 400, 391, 232, 488, 535, 401, 235; 348/419; 711/151

[56] References Cited

U.S. PATENT DOCUMENTS

5,136,582	8/1992	Firoozmand	370/400
5,159,447	10/1992	Haskell et al.	348/419
5,210,479	5/1993	Firoozmand	348/419
5,257,258	10/1993	Birman et al.	370/235
5,315,580	5/1994	Phaal	370/232
5,398,241	3/1995	Witchey	370/391
5,434,872	7/1995	Petersen et al.	714/811
5,859,980	1/1999	Kalkunte	709/232

FOREIGN PATENT DOCUMENTS

0577115 1/1994 European Pat. Off. .

OTHER PUBLICATIONS

AMD, AM79C970 PCnet™-PCI Single-Chip Ethernet Controller for PCI Local Bus, Jun. 1994, pp. 1-868—1-1033.

Primary Examiner—Daniel H. Pan

A network interface transmits data packets between a host computer and a network and includes a first in first out (FIFO) buffer memory with an adaptive transmit start point determined for each data packet. The network interface receives data packets from the host computer via a peripheral component interconnect (PCI) bus. A FIFO control determines the byte length of each data packet based on the header information contained in the first few received bytes of the packet. The FIFO control also measures a minimum fill time indicating the time necessary to fill the FIFO buffer memory with a predetermined minimum amount of data necessary before transmission by the FIFO buffer memory. The FIFO control calculates the time to fill the FIFO buffer memory with each packet based on the determined length and the measured minimum fill time. The time to empty the packet from the FIFO buffer memory is also calculated based upon the length of the packet and predetermined network transmission rates. If the time to empty the packet from the FIFO buffer memory is greater than or equal to the time to fill the FIFO buffer memory, the transmit start point is set to the predetermined minimum amount; otherwise, the transmit start point is adjusted in accordance with the difference in time between filling and emptying the FIFO buffer memory with the packet, a FIFO fill rate based on the measured minimum fill time, and a coefficient that accounts for latencies in the PCI bus.

17 Claims, 4 Drawing Sheets